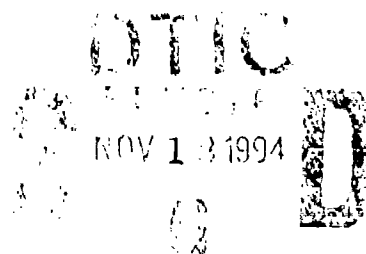# An Assessment Of European Parallel Software Research

Final Report for Travel Grant

submitted to:

Dr. Andre van Tilborg
Computer Science Division
Office of Naval Research

by:

Gul Agha
Department of Computer Science
University of Illinois
Urbana, Illinois 61801

October 25, 1994

94 1117 001

Contents

per ltr

A-1

SUMMARY

This report summarizes research reviewed during trips to a number of institutions in France, Switzerland and Italy to review the state of research in parallel and distributed software. The report makes note of the research models, current status and organizational difference between the U.S. and these countries. As with any such report, it does not aim to be complete; instead it describes the overall organization and focuses on some salient research efforts. The report is most complete about France but includes some institutions in Switzerland and Italy.

Although not comprehensive, certain themes emerge from the study. Among these are:

- There is a strong interplay between industry, government and the computer science research community. European industry is an eager player in University research, in many cases uptaking technology directly, rather than relying on recruiting graduates. Industry often provides real proprietary data, not just funding.

- Governmental agencies are actively supporting applied research in national laboratories. Researchers in such laboratories work closely with University laboratories – to the extent of routinely supervising PhD thesis research and teaching advanced seminars.

- The research often emphasizes a greater interplay of theory and system building. Semantics is considered an important area of theory and people building systems often do formal reasoning for critical parts of the code. (In particular, France has a number of deployed safety-critical software controlled systems, including the autopilot in the AirBus and an operator-less subway line in Lyon.)

1 University of Grenoble

Grenoble is a University town and site of a technology park. The small town surrounded by Alpine mountains acquired its status when it was selected as a nucleus of French nuclear research. The University is situated in a single large campus which houses different Colleges (faculties as they are called) including Law, Medicine, Engineering, and Arts. This is not common in France – the disciplines tend to be segregated and physically separated.

Computer Science research is carried out under the umbrella of IMAG, Professor Traian Muntean is director of research at IMAG-LGI laboratory at the University of Grenoble. (IMAG is an umbrella organization coordinating a number of computer science laboratories in the area.) Muntean's group is doing research in massively parallel programming and architectures. A key accomplishment of this group was the development of the Supernode: a reconfigurable communication architecture for the transputer. More recent research has developed a parallel object oriented operating system Paros which supports object placement and migration, a network routing algorithm for distributed memory machines, and innovative applications of parallel genetic algorithms. These applications include determining process to processor mapping and supporting robot motion planning.

2 University of Nice

The University of Nice has an active Department of Computer Science, located in a new building; the department is slated for growth. Nice is an active center of CS research in France with three different institutions. Besides the University, there is a center for applied mathematics and INRIA. The long range plan is to consolidate various research laboratories in a single area. The teaching load at French Universities can be relatively high – equivalent of 2 courses a semester but there are exceptions to this at selected schools. Moreover, some researchers affiliated with the Universities, may have their primary appointments not from the University (i.e. through the Ministry of Education), but through agencies of the Ministry of Science and Technology.

At the University, several young faculty are establishing a program of research in parallel computing. Among these are Francois Baude, who did her thesis at the University of Paris, Orsay, and post-doctoral work at Cambridge. Her dissertation developed a parallel complexity model for actors and established its relation to the conventional PRAM model. She also showed that for a large class of problems, namely those that are actually data parallel computation, the actor model of complexity was

more realistic. Her interests remain more on theoretical aspects of parallel computing – specifically, on complexity and algorithms.

Professor Dennis Caromel leads an active research group in concurrent object oriented programming. He has designed and implemented a parallel version of Eiffel – extending it by defining special classes to model processes with object-style data abstraction. However, the communication model is closer to actors than that in Communicating Sequential Processes in that messages are automatically buffered. Inheritance is extensively used for modelling in this framework. The root class is a *live routine* from which a set of services can be defined. Requests are stored in a mail buffer which is explicitly checked by a live routine object. Local synchronization constraints are modelled by explicit testing of the incoming message. Moreover, by making the mail buffer explicit, explicit manipulation of the scheduling algorithm becomes possible, e.g., the messages may be processed in first-in first-out, oldest of a set, with real-time constraints, etc. A library of services has been developed but the library is not optimized.

Caromel has also worked on *Electre*, a domain specific language for programming robots. The language supports reactive programming. Although the communication model is asynchronous, with the sender waiting only by necessity, the receiver is interrupted by messages. Thus, unlike parallel Eiffel, putting messages into the queue is explicit in Electre. Wait by necessity is implemented using futures, where calls block if the value to be returned is explicitly required.

The research has not focused entirely on language design and implementation: a semantics for Eiffel has been defined using *Centaur* (see below). Centaur allows the *natural* semantics of a language to be formally expressed; the user specifies how the syntax is to be parsed into an Abstract Syntax Tree and the operational semantics in terms of this representation. Centaur can then be used to support automatic program transformation. A current goal of the Centaur project is to link the operational specifications of programming languages to automated proof tools.

3   CMA: The Center for Applied Mathematics

CMA is an active research center. Researchers affiliated with CMA do not teach undergraduates but may supervise PhD students. The center has an active CS contingent which emphasizes formal rigor in application driven problems. In particular, a large active research group, headed by Gerard Berry, does research on models of distributed real-time systems. The effort dates back to the early 80's and is based on a formal model of concurrency, called *ESTEREL*, suitable for certain real-time systems.

The model assumes a synchronous global clock and uses broadcasting. The broadcasting does not guarantee that a potential receiver will actually be listening – i.e., a listener may miss the broadcast. Thus the communication model is closer to hand-raising than handshaking. Moreover, a broadcast may lead to another broadcast "at the same time."

It is well understood that it is possible to express programs in this framework which violate causality (and thus are impossible to implement). However, the group has developed techniques to statically detect causal violations. A source of the problem is the assumption of no delay between a condition and an action. Berry points out that this assumption, while it seems physically wrong, provides a suitable abstraction for a number of systems such as circuit design. In fact, a high performance data parallel processor, PeRLe, was designed by the DEC Paris lab, using Esterel for specification. Moreover, tools have been developed to support Esterel interface to a verifier and support compositional reduction. Thus the abstract model used by this group has been shown to be of practical use for certain classes of applications.

4   INRIA, Sophia Antipolis and Rocquencourt

The goal of Institut National de Recherche en Informatique et en Automatique (IN-RIA) is to encourage applied research in CS. There are five campuses (besides Sophia Antipolis and Rocquencourt, the other two are in Nancy-Metz, Rennes, and a new one in Grenoble). INRIA has more than 90 industrial and academic partners in France and another 100 in the rest of Europe. INRIA has about 300 permanent full-time researchers. It also hosts 660 other scientists, including about 50 from industry, 180 from other public laboratories, 330 PhD thesis students and 100 invited foreign researchers. High level languages and parallel systems are two active areas of research.

At Sophia Antipolis (near Nice), one of the more influential projects is headed by Gilles Kahn. The goal is to provide a powerful tool, called *CENTAUR*, which allows a formal executable semantics of a programming notation to be specified. In Centaur, the dynamic and static semantics are defined using type rules. A parser is built using automatic tools and structure editors can be applied to the concrete syntax. The abstract syntax tree of expressions (in the specification language) can be visualized and manipulated through customizable menus. The semantics of a number of languages, including SISAL and ESTEREL, have been defined using Centaur. The definitions allow direct implementation of an interpreter.

Centaur is written in LuLisp (a French standard Lisp also developed at INRIA) and Prolog. Thus the implementation is very much actor-like, with functional net-

works responding to asynchronous calls (i.e., message propagation to reactive components) but the current implementation is centralized.

At Rocquencourt, there are a number of active groups. One headed by Marc Shapiro works on distributed operating systems. Another is active in programming language research. Headed by Christian Queinnac, the group is developing Lisp based languages for concurrency. The research has been strong in building a semantics which improves compilation. Queinnec has defined Crystal Scheme which allows parallel evaluation of arguments, and provided a semantic interpretation of continuations to support concurrency. They have also extended Scheme by adding a choice operator which can be used to model schedulers. The research is theoretically sound and provides clear insights into the nature of various linguistic constructs. The use of a powerful sequential language to build on has its advantages but also has some drawbacks. In particular, the complexity of shared namespaces in sequential environments is carried over to the parallel systems, making it much more difficult to do modular reasoning over components.

5 ENSAE

Toulouse is another major center of education and research in France. Among the institutions located here is the elite School of Aeronautical Science, ENSAE (one of several small selective technical colleges reflecting a system started by Napolean to support scientific education). Primarily an "advanced" undergraduate institution, ENSAE is supported by the French Defense Ministry. Students are selected on the basis of a highly competitive examina ¬ n given after the first two years of University level study. A hundred are selected among thousands of applicants. The study emphasizes basic sciences and mathematics and is considered extremely rigorous.

Students in their final year at ENSAE work on independent projects which although equivalent to a Master's thesis in the U.S., are of much higher quality than typical in the U.S. In particular, there is a strong emphasis on group research as well as interdisciplinary (within Engineering) work. Students are trained in a diverse Engineering background, similar to Caltech's Engineering Science major. There is a strong emphasis on actually building prototypes and the teaching of software is integrated with that of hardware. Part of the motivation is the importance of embedded systems to aeronautics.

Some impressive outcomes of research include the development of software for numerical simulation of aviation systems (sold to Boeing) on a client-server architecture. About a year ago, a group of students designed and built a communication satellite

as a special independent project. The satellite was launched by France and worked successfully to provide Amateur radio communication. In another special project, students designed and built an aircraft. They could not test the aircraft, however, due to liability reasons. Needless to say the school is well funded. As usual in France, there is no tuition charge. Moreover, the living expenses of students are covered by a government stipend. The school also receives generous support from the French aviation and space industry.

The faculty at ENSAE work closely with CERT in research projects (see below). Graduate seminars are also held in cooperation with other Universities in the area. Occasionally, students are encouraged to go abroad for their thesis projects (one visited the author's group at the University of Illinois for a semester). Their performance has been impressive.

6 CERT

CERT focuses on research and is linked to both ENSAE and ONERA (another federal agency). ONERA was initially a government division but is now "more" private. ONERA is a superstructure over three similar laboratories, the two others being in Lille and in Chatellion. CERT employees are, however, still treated "as if" they were civil service employees.

The central emphasis of the laboratory is to foster a greater partnership between industry and research universities. CERT hosts PhD students and also has defense contracts. Among the areas of research that are emphasized are CCC systems and numeric (scientific) computation. In particular, the idea is to support autonomous robots in a possibly hostile terrain. Thus, the components of the model include environment interpretation, perception, decision analysis, and command. Twenty researchers work in the perception program and five researchers are working on symbolic fusion using multiagent techniques. (Multiagent architectures are a very active area of research in France. In particular, a number of actor groups have applied their tools to build multiagent architectures, as discussed later in this report. Reflecting the activity in this area, there is a regular French conference on multiagent systems). A formal model of situation assessment has been developed to allow symbolic manipulation and incremental modification of situation representation to account for new information. The agent architecture supports pattern recognition and situation assessment. Multiagent techniques are used to support symbolic fusion and cooperative decision making.

Another major project called SATURNE, now complete, built a distributed computer architecture.

7 IRIT, Toulouse

There is a major computer science department at Toulouse. Researchers in CS at the University are part of IRIT which is a superstructure for coordinating CS research at Paul Sabatier University and at ENS in Toulouse (see ENS in Lyon). Paul Sabatier University has an active PhD program and is situated on a modern building at the technology campus. IRIT has an active group doing research in theoretical aspects of programming under the direction of Patrick Salle. Besides Salle, there are three other permanent persons associated with the project – some of them long-time research faculty – and a number of graduate students.

The group has been focused on actor research since 1979. The group implemented an early actor language, based on functional constructs, called Plasma-1. The language supported full pattern matching, continuations, and used no side-effects. However, it was interpreted and sequential. In a sense, it was similar to Scheme (also inspired by early research on actors). Research based on Plasma-1 ended in 1984. However, the French have developed LuLisp, which serves as a virtual standard and uses a general virtual machine approach similar to that in Plasma.

In the mid-eighties, the programming language group developed Plasma-II which is based on the pure Agha model of actors. Plasma-II is parallel, provides explicit serialization (thus modelling local state change). Plasma-II provides a number of language constructs which are implemented in terms of the pure actor model. Such constructs include blocking communication, futures, broadcasting, and data parallelism through parallel list evaluation. Actor behaviors are modelled as closures with links to the environment and a copy of their script. The links to the environment represent a list of continuations and the explicit representation of script allows behaviors to be decomposed and built (an actor script is the abstract syntax representation of an actor's behavior definition). Plasma-II supports Lisp like reflective function *eval*. The implementation is fully parallel although the executables mix parallel and sequential constructs. The execution model uses n elementary virtual machines (which may be mapped to physical processors). Each EVM controls time-sharing explicitly. Plasma-II has been implemented on networks of workstations, transputers, and a butterfly network. A number of tools to support programming have been developed. These include a debugger, a compiler for sequential target code, and support for persistence of stable actors (i.e. those in quiescent state). The debugging support is divided into the two graphical debuggers – a debugger controlling execution on the EVM's and a Plasma-II level debugger which provides traces, breakpoints and post-mortem event diagram analysis for remote effects. The debuggers use Suntools and Xwindows.

Since 1989, research has focused on three areas. One area of research has been optimizing the distribution of actors for efficiency, i.e., issues related to distributed ob-

ject management. These have interacted with topics in compilation and the study of applications. Among the concepts developed are dynamic and automatic actor placement, static analysis and rewriting (adding traces) to optimize placement through automatic migration, and message flow analysis using test sessions to compute minimum cost distributions.

A second area of research has been Parallel ADTs. The idea is to provide ML-like ADTs for concurrent computation. Parallel ADTs are build out of (distributed) actors and inductive parallel operators. Thus type constructors may be used to build scripts and operator applications may be defined in terms of parallel message broadcasts using *superfunctional* iterators. The third area of research is in reflection. Besides the use of application for dynamic control of tracing and placement, reflection has been used to build adaptive actor programs using modification of scripts. The reflection is based on a one-to-one mapping between meta-actors and actors. It does not use a group construct (unlike Yonezawa's research at the University of Tokyo, and Tokoro's at Keio). Moreover, the reflective model does not reflect over mail queues – a f..rm found useful by the author's group for fault-tolerance applications. However, unlike the other efforts, research by Salle's group shows the power of manipulating the control flow by making continuations explicit as well as the effect of manipulating scripts.

Future work will focus on distributed A.I. applications and use of CENTAUR (see INRIA Nice above) to define a "natural" semantics for Plasma-II.

Salle's group is well-known throughout France and has had a significant impact on the research in actors, distributed agents, and programming languages in France.

8 University Claude Bernard, Lyon

Lyon is another important center of research. Lyon is the second largest city in France (about an hour away from Grenoble, another important technology center in France).

A number of research groups in Lyon are organized under the umbrella of Laboratoire d'Ingenierie des Systems d'Information (LISI). The participants in LISI are the University of Lyon and the National Institute of Applied Sciences (INSA). Co-operation with the INSERM (the equivalent of NIH in the U.S.) also takes place in the study of medical informatics – areas such as pattern recognition, signal theory, and medical data processing. Four of the primary groups that are part of LISI are doing research in areas of distributed Artificial Intelligence, multimedia systems and and distributed knowledge and databases. A new group established at the University

(at the Mathematics department) is doing research on parallel numerical computation. Other institutions active in CS research in the area include LIP (Laboratory for Parallel Informatics) at the Ecole Normale Superiure, Archipel, an ESPRIT funded lab building transputer based parallel computers, and a small CS department at the Ecole Centrale with one professor doing research in GUI's. There is also a national supercomputing center in Lyon, one of the first in France.

Professor Jean-Marc Fouet heads the group working on Meta-knowledge architecture. Dr. Salima Hassas, who did her thesis on a reflective model of actors for use in distributed Artificial Intelligence systems is now a member of the group. The group has developed on the Gosseyn Machine for problem solving. The goal is to organize and compile knowledge in terms of distributed systems. Actors are used as an intermediate level model for distributed computing. Production rules and first-order logic are used as the top level representation. This representation is translated into actors which are implemented using low level dataflow processors. The system acquires knowledge, compiles rules, and improves its organization through adaptation. Genetic algorithms are used for adaptation. The language used is Lisp based. Local rule-based systems are translated into the actor language (using incremental compilation) and case based learning is used. The application areas have been discovering molecular structure in chemistry (a project in cooperation with the Chemistry department) and perception. For example, constraints are deduced from spectral data, e.g. distance constraints between Carbon atoms. The system then tries to generate graphs to satisfy the constraints; sometimes there are thousands of solutions some of which may be isomorphic and some of whose images may need to be resolved by removing noise. The constraints are represented by networks of rules.

Gosseyn uses a pure actor implementation. The actor model is reflective and meta-rules are provided to solve internal problems. Distributed objects provide different expertise, e.g. for placement. Meta-rules resolve internal structures, e.g., when a rule is added the system checks for consistency of new rules and the logical relation between the rule and the existing system (conditional relations). In case of inconsistencies, non-monotonic reasoning is used. Traces are analysed to cache information and preferences in user interface are detected based on patterns of use.

9 University of Milan

Milan, the commercial capital of Italy is also the site of two Universities. The University of Milan focuses on CS whereas the Engineering University hosts Computer Engineering. Research at the University includes a number of groups working on parallel and distributed computing. Among the topics are formal specification languages

and concurrent object oriented programming.

Fiorella De Cendio has worked on a net model of inheritance in a framework unifying process algebras and net theory. The key result is to add algebraic specifications to nets. The framework uses synchronous communication in a specification language called CLOWN (Class Oriented with Nets). The body of the process is a net. A key semantic issue is how to preserve the equivalence substitutability of the semantics in the presence of inheritance. De Cendio achieves the desired substitutability by restricting inheritance to preserve a specific pre-order (called $ST_{\leq}$)). Essentially, behavior inheritance in this case is behavior extension rather than behavior overwriting: the algebraic signature of the parent must be less than that of the descendant. In other words, the precondition may be weakened and the post-condition strengthened when inheriting. An application domain is provided by Electric Power Company which has also provided data under a nondisclosure agreement.

A second group, headed by Professor Francesco Tisato, works in the real-time concurrent systems. The specific goal of the effort is to build an object-oriented class library without embedding Real-Time constraints into the classes. They currently have prototype versions implemented. The model of real-time computation is descended from actors and implemented using C++. Methods are uninterruptable atomic actions, and the communication mechanism is asynchronous, following the actor model. Other communication protocols are specified using specific actors called connectors. If a timing constraint is embedded in the message, then it cannot be redefined. The idea is to separate the logical specification of an actor's behavior from the timing constraints imposed on that behavior. A real-time event driven planner is connected to an actor (possibly a connector). Pending events are scheduled using a virtual clock time.

Their experience suggests the lack of pre-emption of methods is critical in simplifying scheduling. Asynchronous (send no-wait) communication is necessary in distributed systems to keep method execution duration short enough to avoid necessitating preemption. The thesis of the research, that schedule specifications can be transformed into executable code, has not yet been demonstrated in their framework. Before using actors, the group implemented an approximation in ADA (providing ADA tasks for actors). However, preemption in ADA made scheduling impossible. The research is funded by the European Space Agency and by a personal communication appliance manufacturer.

10   University of Salerno

Salerno was the site of the oldest medical university in Europe. The University of Salerno supports a campus in a picturesque area south of Naples. The university has two CS related departments, one in Computer Engineering and one in Computer Science. Both have researchers using the actor model. A research group headed by M. Di Santo is working on language support for parallel computing. An actor system kernel (ASK) has been implemented; the kernel provides basic actor primitives on transputers. The kernel is written in C. The group is using the kernel to develop systems for computer vision; the goal is to develop a complete library of parallel object-oriented routines for computer vision. G. Ianello, who worked with the group at Salerno, has moved to the University of Naples and is continuing research on implementations of actor languages and parallel architectures. P. Cattaneo is another researcher working on developing programming environments for actors.

11   Ecole Polytechnique de Lusanne

There are two preeminent institutions for Science and Engineering in Switzerland – EPL in Lusanne and ETH in Zurich. The institutions are known as Swiss Federal Institutes of Technology. Computer Science research in EPL is divided into 9 laboratories. Three of these are particularly relevant to parallel and distributed computing: they work on distributed systems, Teleinformatics (computer communication, e.g. ATM's), and programming languages.

Professor Andre' Shiper heads the laboratory in distributed computing. The laboratory has nine people, four working on languages and representation and five on algorithms. PhD thesis students are counted as full-time researchers. The language and system research subgroup is headed by Rachid Guerroni. The distributed computing effort is focused on distributed fault-tolerant systems with, naturally, banking in mind. There is also support from the Swiss NSF equivalent (FNRS) and collaboration with ESPRIT.

Professor Shiper was involved in research on ISIS at Cornell and there remains an active relationship between Cornell and ETH involving student exchanges. Moreover, ISIS is in use by some one hundred odd banks in Switzerland. ISIS is a distributed operating system developed at Cornell, which uses a virtual synchrony model. The systems subgroup has used a Smalltalk based language on top of ISIS to provide replication, concurrency control, etc. Specific objects, called encapsulators, are mapped to ISIS and higher level class representations are provided for causal broadcast (cbcast) and abcast (atomic multicast). Smalltalk messages are transformed by abcast

or cbcast. The transformation is done by messages that are intercepted in the encapsulators around objects; thus, it is transparent to the application. System calls are made to ISIS which runs on top of UNIX. A collaboration with a private company, Landis and Gyr, also provides applications to Computer Integrated Manufacturing.

Professor Claude Petit Pierre heads a laboratory in Teleinformatics. The laboratory has worked on both hardware and software. For example, they collaborated with AT & T to design a switching system. The group has moved in recent years from using C to C++. A concurrent version of C++ has been developed by the group. It adds synchronization to C++. FIFO buffers are assumed and signals are used for flow control. Protocols are layered and CCS is used for validation (by making the message buffers explicit, synchronous blocking calls to other objects may be used to build other communication protocols). However, a weakness of this model is that synchronization constraints cannot be incrementally modified. The model of object used is quite complex: an object may have internal concurrency; time sharing is used between two bodies within an object.

Professor Charles Rapin heads a small laboratory in programming languages and compilation. The group uses a programming language called Newton, which was developed locally in the early eighties as a pedagogical exercise. Newton is class-based, single inheritance language with higher-order functions and type polymorphism. Ordinary objects are allocated on the heap; there are also co-routines and active objects. The implementation is quasi-parallel (uses time-sharing). Current effort is to distribute objects onto a network and include a number of parallel constructs such as future variables, monitors, and multicast group. The language is quite large, and is becoming larger still. Active objects are reactive, but may explicitly buffer messages. When a new message arrives, it interrupts an object and the old context is saved. This can create consistency problems as the new message may modify the state (by contrast, actors separate local state changes caused by messages by letting messages lock an actor until the new local state is constructed).

12 Geneva University

Geneva University has a small department of Computer Science in a campus near the center of town. The department has a large active group in concurrent object oriented programming under the direction of Professor Tsichritis who is currently on leave and working as head of GMD in Bonn. The group has applied COOP programming techniques for a number of distributed applications, such as computer music and multimedia. Active objects are used to wrap pieces of hardware, such as a CD player or a video player, and software. Video actors perform various functions, for

example, a video assistant may explain how to use a fax machines in a demo mode. A large demonstration of the system is a virtual museum built using video widgets. The project has involved about 5 person-years to build the prototype, with funding from DEC and FNRS (the Swiss NSF). Earlier ESPRIT funded an effort to construct a software library.

Prof. Bastian Chopard heads a group in Parallel Computing. There are 3 other members in the group. The laboratory has a 32 node transputer, a 8000 processor CM 2, and an IBM SP1. The transputer is used primarily in parallel architecture courses since physicists don't want to use OCCAM. The CM2 use on real applica.ions is higher because there are more tools available with it. T..e laboratory has an ongoing collaboration with the physics department. Problems being addressed include modelling diffusion systems (n-body problem), lattice gas model (non-equilibrium systems), and discrete simulations.

Another effort, supported by Swiss Telecom, is studying the wave propagation in Geneva in order to optimize the placement of receivers to maximize signal strength for cellular phones. Specifically, reflection and absorption by buildings affects the signal strength. Data parallelism is used: each pixel is mapped to a different processor and the Geneva map and building structure is input. Each simulation of 1024 x 1024 pixel size takes a few seconds. Boundary conditions are not modeled.

Other parallel machines in Switzerland include an NEC 64 (large vector machine) at the CSCS (National Center for Scientific Computation) in Lugarno and a Cray at Lusanne. The FNRS is funding the parallel scientific computing. There is also industrial funding for applications of massive parallelism to financial data analysis. The genetic algorithm is used to optimize parameters. Proprietary data are provided by financial institutions under non-disclosure agreements with the understanding that the methodology, but not the actual data, may be published.

Prof. Harms heads the Computer Communication Group. The group has three researchers working on network management for distributed applications and three others working on communications architecture to serve as a successor to OSI and TCP/IP. One goal of the communication architecture project is to make electronic mail more reliable. Another is to determine mechanisms for performing activities without the same protocol on both sides messengers. One scheme carries program scripts to remote site —- allowing protocol interpretation (sort of like communication by virus).

A protocol code is expressed in a *comscript*, an enhanced version of postscript. Comscripts are event driven and create processes. A parent process can reconfigure child processes. The process stack is reconfigurable, allowing removal of layers for customizability. For example, to increase transmission speed *IP* can be removed

from TCP/IP. An issue being investigated is what minimum functionality must be present to allow such dynamic replacement of layers. The key idea is to provide protocol wrappers which allow layers to be flushed gracefully. The work has important implications for distributed O.S., allowing the use of *virtual* microkernels — which allow change of protocols between microkernels. (This is similar to the direction of the author's work in providing transparent dynamic fault-tolerance).

A key concern in the network management for distributed applications is interoperability. The effort is building on a Concurrent Object Oriented Framework based on COOL (Concurrent O-O language). The laboratory is also critically evaluating the CORBA standard proposed by Object Management Group. Specifically, they are studying issues such as whether migration should be real or simulated by creating proxies. The issues are studied in collaboration with several industrial contractors, e.g. Siemens Nixdorf (which is a member of OMG) and ABB (a Swiss Multinational manufacturer).

The ABB contract aims to provide a federated database with integrated CAD/CAM support. For example, the ABB has 15 different stages from planning to after sale support, each modelled independently with different databases on different machines, and running into millions of lines of code. Security is a particular concern as information is compartmentalized and may not be exposed. ABB is providing obsolete data for modelling purposes. The researchers at the Geneva lab are collaborating with University of Zurich in building a prototype in Hybrid, a concurrent object-oriented language developed in Tzichritis group by Oscar Nierstrasz (now a faculty member at Bern University).

13 CERN

There is a small computational science group at CERN to support computational physics for experiments conducted there. The primary parallel computation technique used is *event farming* on networks of workstations. A parallel machine in use is a 32 node Meiko (transputer based architecture). They are moving to MPI and Fortran 90. There is some debate about moving to C++. However, the CERN programming library is about 2 million lines of Fortran code, so there is the usual concern about legacy code.

14   University of Paris, Orsay

The University of Paris at Orsay had a small active group under the direction of Professor Vidal-Nacquet. While on leave from Orsay, Vidal-Nacquet spent three years heading a large industrial R & D group at Alcatel-Alsthom. At Orsay, his group has done considerable research in concurrency and specifically actor systems. In particular, a PhD thesis on actors developed a parallel computational complexity model of actors, and another defined a semantic model based on higher-order nets. More recent research at Orsay has focused on algebraic specification of concurrent programs. Vidal-Nacquet is also affiliated with Supelec (another selective school like ENSAE described above). His group has combined synchronous concurrency (as in Esterel, described above) with the actor approach. In particular, they have developed tools which enable a C++ object to be considered a synchronous module (as in Esterel, see above description of CMA) . Composition, instantiation and inheritance of such objects are supported.

15   University of Paris VI

Sarbonne, in the center of Paris, is the site of one of the oldest Universities in Europe. Paris VI is the successor technical university, housed in a sprawling complex not far from the old ruins. There are several laboratories at the University doing research related to concurrency. They are organized as part of the Institut Blaise Pascal. In particular, there are two groups of researchers in parallel and distributed research.

Jean-Francois Perrot is the leader of a group which has built a concurrent programming language Actalk, based on Actors. the language is implemented in Smalltalk and has been used extensively for teaching actors. Moreover, Actalk is extensible and provides a testbed for modeling, classifying and experimenting with object-oriented concurrent programming (OOCP) languages. Since its initial design during mid-88, Actalk has been used to study topics in object-oriented concurrency including reflection and exception handling. Actalk provides a number of strategies for synchronization and implicit reply. The implementation has been widely available and extensively used in France and abroad. (One of the architects of Actalk, Dr. Jean-Pierre Briot, is currently an international collaborator on the MITI sponsored Japanese national project on Real World Computing). Professor Jacques Ferber's laboratory, doing research in distributed AI in IBP, has used Actalk to implement a library for constructing multi-agent systems and applied to various domains.